Kevin Martin – CIS700 Monday @ 8:00pm EST

4/27/2020

Deliverable 1

**Overall Goals/Research Hypothesis**

The dataset1 I will be using shows the historical data for various cryptocurrencies from January 1, 2016 - April 17, 2020. The data includes many different cryptocurrencies (as opposed to just Bitcoin), as well as multiple trading metrics besides price. There are many, many records to go through, and 17 available columns. These fields are both numerical and categorical, exhibiting nominal, ordinal, interval, and ratio qualities.

Each row represents an entry for a specific currency on a specific day. So there could be multiple entries on the same day. The rest of the fields help describe the nature of the transaction, as well as provided qualitative information on that specific security.

The stated research goals for the data are described below:

1. Which features of cryptocurrency trading will best predict future price movements, and can we we further predict the magnitude of such movements?
2. Given the variety of features across time, can we find an optimal combination to predict the future price movements?

**Previous/Related Contributions**

There are two previously started kernels on Kaggle for this dataset. The first appears to be a pretty basic start in learning about the data, while the second is a more robust attempt to derive some meaning from the data.

1. https://www.kaggle.com/georgezakharov/kernel1f75d2a722

The author provides some details of his workbook, but ultimately not much in the way of any real data analysis. Kaggle will generally provide a "starter" kernel for much of their data, and to standardize this, it is packaged in a Docker container. The references to Docker remain in the notes to the code, so it appears that the beginning steps are generic.

The author examines various properties of the data using the pandas libarary in Python. It's helpful to see the different data types presented, as well as some common statistics. However, the data has not been cleaned or formatted at all, and the aggregations do not help much. For example, "df.describe(include='all') is an extremly quick and easy way to see key summary statistics. Unfortunately it requires that missing values be removed or excluded and will return an error in most cases. Here we can see that the data was left completely untouched, resulting in many "NaN" errors. On the other hand, this is good to be aware of early in the process as I now know that a deeper dive is required.

Overall this is a small first step and appears to be more of a placeholder. Some of these techniques I will also be utilizing, but again, just to get the most basic of understnadings before proceeding into more reobust analysis.

1. https://www.kaggle.com/abasov/bitcoin-price-prediction-with-keras

In contrast to the previous kernel, here we see almost a completed notebook. While there is little in terms of descriptive notes, the author's code is well documented and easy to follow.

The dataset includes metrics on a multitude of cryptocurrencies, however here the author chose to just focus on one: Bitcoin. In doing so, the noise of less popular or complete cryptocurrencies is removed, and a more targeted approach can be taken. All the data checks and review revolve around a single feature. So in this way, errors are more easily detected and data becomes more manageable.

After an initial pass through the data, including helpful charts and summary metrics, the author provides the code to begin training his model. Beginning in output line 12, the tuning begins and the author leaves comments to help walk through the approach. Here the goal is to find the optimal timeframe, or epoch, that gives the most accurate predictions. Because the data is ordered by time, this makes for a relatively easy setup. In looking at the graphs provided, one can see the fit of each epoch's results in comparsion to the real price of Bitcoin on that day.

In looking more closely at the libraries imported (input line 1) and the model generation (input line 11), we can see the author implemented a long short-term memory (LSTM) approach using Keras. As we have discussed that the data is across multiple dates, this time series prediction method makes sense. By training the model with multiple historic time frames, he can optimize the model and apply it to the unseen data and test the ultimate success.

While the author does not porvide a formal summary of his findings, he does present a final result on input line 14. At this point he simply layers the real price over the predicted price, presumably with an epoch of 15. The similarities are displayed visually in the last graph. It appears, at least to the eye, a fairly close predictor. However summary table with more precise notes would certainly be helpful. It is difficult to estimate or comment on the success of this model based on this last graph, but the process to arrive here does at least seem thorough.

**Feature Selection/Engineering**

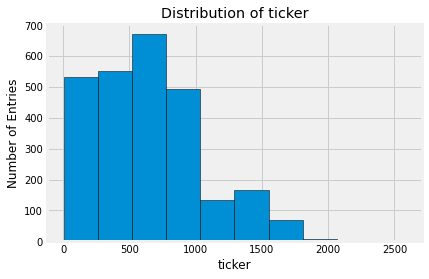
The dataset is in decent shape, but it is missing a lot of entries for what appear to be some of the more "obscure" crytpocurrencies (also referred to as coins). As such, I will be removing those entries which do not have any labels.

There is plenty of data still for the major/complete entries, and for these purposes I want to focus as much as possible on tangible transactions. Unfortunately, missing data is a bit of a problem. To address this, we will look first to the attributes themselves to make sure they are useful. Any field with less than 50% of the records entered will be removed.

Second, we will look across the records. The field for crytpo\_name should be complete, and if they are missing, we will remove them. The major cryptocurrencies (such as Bitcoin) provide this field, and thus will be kept.

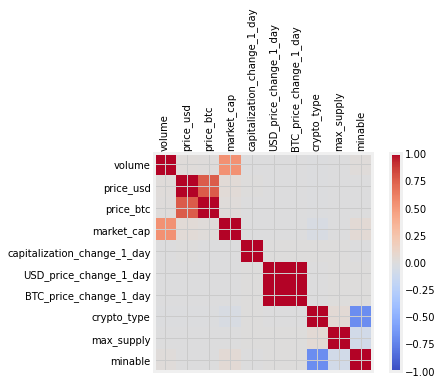
Even after removing the coins which weren't labeled, we are still left with over 2,000 different currencies. So I'll dive a litle further and look at a histogram of how many times each coin appears.

The results are are quite varied: just under 700 on the high end and next to nothing on the low end. Also for clarity I've removed all 2,000+ of the currency names.



*Figure 1*

To get a general sense of corrleations, a heatmap across all vairables as compared to all others was generated. Clearly, there are some moderatey correlated attributes, but to the point of attrition, a more focused list of correlations was generated. Here, we can see the more relevant relationshiops and begin to map out where the focus of the analysis will be directed.

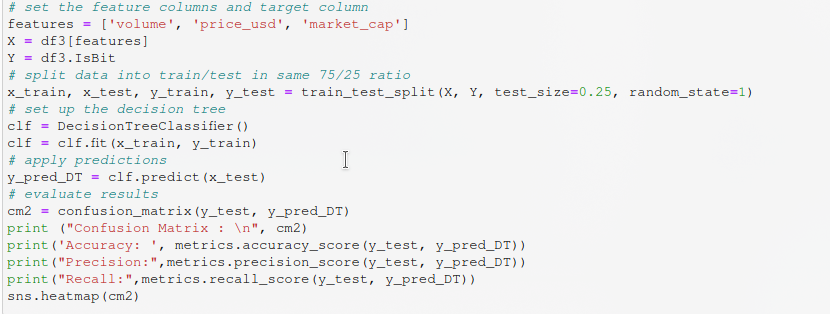


*Figure 2*

The key data metrics are going to be the currency identifier, price in USD, volume, market cap, and the date. From those fields we will be able to derive lots of useful and pertinent information. Additionally, the binary field of if a coin is mineable or not is also an interesting one. Thus far I have not looked at it in depth, but I believe it may provide additional color into some of the analytics that are to be performed.

Using the results from Assignment 1, I was able to get an accurate classification model set up on the data. Basically, I wanted to see if the trading data given (price and volume in particular) were sufficient enough to identify if a transaction was in Bitcoin or in another currency.

Bitcoin is by far the most popular and widely traded, and as a result trades in much higher amounts and more frequently. Thus it should be easy for a model to figure out if a trade was or was not in Bitcoin. The risks would be that all crytpocurrencies are extremly volatile and subject to large price/volume fluxuations. So it’s possible another popular coin traded unusually high for a day and thus throws off the model. However upon analysis, it was very easy for even a lightly trained model to identify correctly. For example, consider the following model which uses the sci-kit learn module for Python. It automatically split the training and testing data for me, and using the stock hyper parameters the results were quite high:

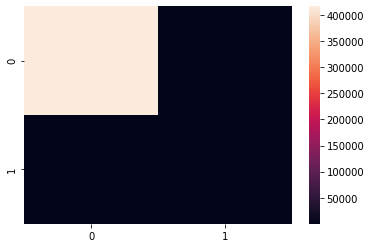


The results of the best performing model decision tree split on 75% training data/25% testing data are as follows:  
  
Accuracy: 0.9999113252088052

Precision: 0.9595238095238096

Recall: 0.9527186761229315

The resulting confusion matrix was also quite telling:



*Figure 3*

This serves as a partial plot for three of the most important features: price, volume, and currency. At this point in the project, I believe that these will be the three most important features, however there are still others to explore. My Assignment 1 includes a more robust explanation of my code, I used that as my starting point for this Milestone.

While interesting, the results do not yet get to the goal of predicting trades prices. Moving forward, I want to investigate the price and volume relationships across coins, and next day moves. Fortunately the dataset has much of this available already.